

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日

Date of Application:

2002年 7月15日

出 願 番 号

Application Number:

特願2002-206202

[ST.10/C]:

[JP2002-206202]

出 願 人

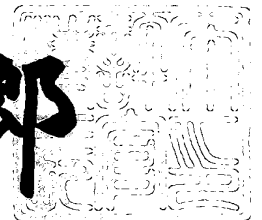
Applicant(s):

インターナショナル・ビジネス・マシーンズ・コーポレーション

2003年 2月 7日

特 許 庁 長 官
Commissioner,
Japan Patent Office

太田 信一郎



出証番号 出証特2003-3005349

【書類名】 特許願

【整理番号】 JP9020093

【提出日】 平成14年 7月15日

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 7/00

【発明者】

【住所又は居所】 神奈川県大和市下鶴間 1 6 2 3 番地 1 4 日本アイ・ピー・エム株式会社 東京基礎研究所内

【氏名】 安部 麻里

【発明者】

【住所又は居所】 神奈川県大和市下鶴間 1 6 2 3 番地 1 4 日本アイ・ピー・エム株式会社 東京基礎研究所内

【氏名】 小柳 光生

【発明者】

【住所又は居所】 神奈川県大和市下鶴間 1 6 2 3 番地 1 4 日本アイ・ピー・エム株式会社 東京基礎研究所内

【氏名】 小野 康一

【発明者】

【住所又は居所】 神奈川県大和市下鶴間 1 6 2 3 番地 1 4 日本アイ・ピー・エム株式会社 東京基礎研究所内

【氏名】 堀 雅洋

【発明者】

【住所又は居所】 神奈川県大和市下鶴間 1 6 2 3 番地 1 4 日本アイ・ピー・エム株式会社 東京基礎研究所内

【氏名】 仲池 卓也

【特許出願人】

【識別番号】 390009531

【氏名又は名称】 インターナショナル・ビジネス・マシーンズ・コーポレーション

【代理人】

【識別番号】 100086243

【弁理士】

【氏名又は名称】 坂口 博

【代理人】

【識別番号】 100091568

【弁理士】

【氏名又は名称】 市位 嘉宏

【代理人】

【識別番号】 100108501

【弁理士】

【氏名又は名称】 上野 剛史

【復代理人】

【識別番号】 100104880

【弁理士】

【氏名又は名称】 古部 次郎

【手数料の表示】

【予納台帳番号】 081504

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9706050

【包括委任状番号】 9704733

【包括委任状番号】 0004480

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 データ処理方法、差分計算方法、これを用いた指示情報生成システム及びプログラム

【特許請求の範囲】

【請求項 1】 コンピュータを制御して、構造化文書の所定の要素を指示するデータ処理方法であって、

所定の指示情報により要素を指示された構造化文書が変更された場合に、当該構造化文書を入力して変更内容を解析し、解析結果をメモリに格納するステップと、

前記メモリから前記解析結果を読み出し、解析された前記変更内容に応じて、変更後の前記構造化文書中の対応する要素を指示するように指示情報を更新するステップと
を含むことを特徴とするデータ処理方法。

【請求項 2】 前記指示情報を更新するステップでは、XPathを用いて記述された前記指示情報を更新することを特徴とする請求項 1 に記載のデータ処理方法。

【請求項 3】 コンピュータを制御して、2つのツリー構造データの差分を計算する差分計算方法であって、

メモリから処理対象である2つのツリー構造データを読み出して比較し、一方のツリー構造データから他方のツリー構造データに変形するための操作をツリー構造の構成要素に対する所定の操作の組合せで表現した操作列のリストを作成し、メモリに格納する第1のステップと、

前記メモリから前記操作列のリストを読み出し、当該操作列のリストのうちで、前記構成要素の移動として把握できる操作列を移動の操作に変更する第2のステップと
を含むことを特徴とする差分計算方法。

【請求項 4】 前記第1のステップでは、前記ツリー構造データを変形するための操作を、ツリー構造のノードまたはサブツリーの追加、削除、変更の組合せで表現した操作列のリストを作成することを特徴とする請求項 3 に記載の差分

計算方法。

【請求項 5】 複数の構造化文書の差分を計算する差分計算部と、

前記差分計算部により計算された差分に関する情報に基づいて所定の構造化文書における所定の部分を指示する指示情報から他の構造化文書中の対応する部分を指示する指示情報を生成する指示情報生成部と
を備えることを特徴とする指示情報生成システム。

【請求項 6】 構造化文書の構造を解析してツリー構造データに変換する文書解析部をさらに備え、

前記差分計算部は、前記文書解析部にて変換された複数の構造化文書に対応する複数のツリー構造データを比較して差分を計算することを特徴とする請求項 5 に記載の指示情報生成システム。

【請求項 7】 前記差分計算部は、ツリー構造データの構成要素のうちで、前記複数のツリー構造データを一方から他方へ変形するための操作において移動された構成要素を追跡するように当該ツリー構造データの差分を計算することを特徴とする請求項 6 に記載の指示情報生成システム。

【請求項 8】 前記指示情報は、X P a t h で記述されていることを特徴とする請求項 5 に記載の指示情報生成システム。

【請求項 9】 前記指示情報生成部は、前記構造化文書の差分と前記所定の構造化文書に対する X P a t h とに基づいて、X P a t h を構成するロケーションステップを再生成することにより、前記他の構造化文書に対する X P a t h を生成することを特徴とする請求項 8 に記載の指示情報生成システム。

【請求項 10】 コンピュータを制御して、構造化文書の所定の要素を指示するデータ処理を実行させるプログラムであって、

所定の指示情報により要素を指示された構造化文書が変更された場合に、当該構造化文書を入力して変更内容を解析し、解析結果をメモリに格納する第 1 の処理と、

前記メモリから前記解析結果を読み出し、解析された前記変更内容に応じて、変更後の前記構造化文書中の対応する要素を指示するように指示情報を更新する第 2 の処理と

を前記コンピュータに実行させることを特徴とするプログラム。

【請求項 1 1】 前記プログラムによる前記第 1 の処理は、
変更前後の前記構造化文書をツリー構造データに変換する処理と、
当該ツリー構造データの差分を計算する処理とを含み、

前記プログラムによる前記第 2 の処理では、前記ツリー構造データの差分を用いて前記指示情報を更新する処理を前記コンピュータに実行させることを特徴とする請求項 1 0 に記載のプログラム。

【請求項 1 2】 前記プログラムによる前記差分を計算する処理では、前記構造化文書の変更に応じて変形されたツリー構造データの構成要素のうちで、当該ツリー構造データの変形に要する操作において移動された構成要素を追跡するように当該ツリー構造データの差分を計算する処理を前記コンピュータに実行させることを特徴とする請求項 1 1 に記載のプログラム。

【請求項 1 3】 前記プログラムによる前記第 2 の処理では、変更前後の前記構造化文書の差分に基づいて、前記指示情報が記述された X P a t h を構成するロケーションステップを再生成することにより、当該 X P a t h の記述を更新する処理を前記コンピュータに実行させることを特徴とする請求項 1 0 に記載のプログラム。

【請求項 1 4】 コンピュータを制御して、2 つのツリー構造データの差分を計算するプログラムであって、

メモリから処理対象である 2 つのツリー構造データを読み出して比較し、一方のツリー構造データから他方のツリー構造データに変形するための操作をツリー構造の構成要素に対する所定の操作の組合せで表現した操作列のリストを作成し、メモリに格納する第 1 の処理と、

前記メモリから前記操作列のリストを読み出し、当該操作列のリストのうちで、前記構成要素の移動として把握できる操作列を移動の操作に変更する第 2 の処理と

を前記コンピュータに実行させることを特徴とするプログラム。

【請求項 1 5】 前記プログラムによる前記第 2 の処理では、前記操作列のリスト中に、前記ツリー構造データにおける同一の構成要素を削除する操作と追

加する操作とが存在する場合に、これらの操作に替えて当該構成要素を移動する操作を当該操作列のリストに加える処理を前記コンピュータに実行させることを特徴とする請求項14に記載のプログラム。

【請求項16】 前記プログラムによる前記第2の処理では、前記操作列のリスト中に、前記ツリー構造データにおける所定の構成要素を変更する操作が存在する場合に、所定の規則に基づいて、当該操作を構成要素の移動を含む別の操作に置き替える処理を前記コンピュータに実行させることを特徴とする請求項14に記載のプログラム。

【請求項17】 コンピュータを制御して構造化文書の所定の要素を指示するデータ処理を実行させるプログラムを、当該コンピュータが読み取り可能に記録した記録媒体において、

前記プログラムは、

所定の指示情報により要素を指示された構造化文書が変更された場合に、当該構造化文書を入力して変更内容を解析し、解析結果をメモリに格納する処理と、

前記メモリから前記解析結果を読み出し、解析された前記変更内容に応じて、変更後の前記構造化文書における対応する要素を指示するように指示情報を更新する処理と

を前記コンピュータに実行させることを特徴とする記録媒体。

【請求項18】 コンピュータを制御して2つのツリー構造データの差分を計算するプログラムを、当該コンピュータが読み取り可能に記録した記録媒体において、

前記プログラムは、

メモリから処理対象である2つのツリー構造データを読み出して比較し、一方のツリー構造データから他方のツリー構造データに変形するための操作をツリー構造の構成要素に対する所定の操作の組合せで表現した操作列のリストを作成し、メモリに格納する処理と、

前記メモリから前記操作列のリストを読み出し、当該操作列のリストのうちで、前記構成要素の移動として把握できる操作列を移動の操作に変更する処理とを前記コンピュータに実行させることを特徴とする記録媒体。

【請求項 19】 所定の HTML 文書に対して設定されたアノテーションデータを管理するアノテーションサーバにおいて、

アノテーションデータが設定された前記 HTML 文書が変更された場合に、変更前後の当該 HTML 文書の差分を計算する差分計算手段と、

前記差分計算手段の計算により得られた差分情報に基づいて、前記アノテーションデータを前記 HTML 文書の所定の部分に対応付ける XPath を更新する XPath 更新手段と

を備えることを特徴とするアノテーションサーバ。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、XML (Extensible Markup Language) や HTML (Hypertext Markup Language) など記述された文書の要素を指示する方法に関し、特に文書が変更された場合に要素の指示表現を更新する方法に関する。

【0002】

【従来の技術】

インターネットなどのデータ交換に用いられる XML や HTML など記述された構造化された文書（以下、構造化文書）に対し、この構造化文書中の特定の要素（エレメント）を指示してアノテーション（annotation：注釈）などのメタ情報を付す場合がある。また、構造化文書に予め変形ルールを記述しておき、この変形ルールに基づいて構造化文書を変形させる場合がある。このようなメタ情報や変形ルールを構造化文書に付加する場合、XPath (XML Path Language) を用いて当該構造化文書における所定の位置を指示し、外部文書から参照することが行われる。

【0003】

XPath は、構造化文書の特定の部分を指し示すための言語であり、XPath を指示情報として利用することによって、構造化文書中におけるアノテーションを付したり変形を行ったりする位置を任意に特定することができる。以下の説明では、XPath による記述も単に XPath と記す。

具体的な X P a t h の表記は、次のように行われる。

図 1 8 は、構造化文書である XML 文書の構成例を示す図である。X P a t h では、ルート要素をスラッシュ「/」で表す。したがって、図 1 8 の XML 文書に対し、要素 a は、ルートの子供要素であるため、「/a」と表現される。また、要素 b や要素 d は、「/a/b」や「/a/b/d」と表される。さらに、「//p[id="foo"]」は、ある XML 文書中に存在する全ての p 要素の中で、その要素の i d 属性が「foo」であるものを選択する X P a t h の表記例である。

【 0 0 0 4 】

【発明が解決しようとする課題】

上述したように、X P a t h によれば、XML 文書や HTML 文書といった構造化文書の特定の要素を任意に指示することができる。しかし、指示対象である構造化文書が変更されてしまうと、構造化文書中の要素やその位置が変わってしまうことにより、X P a t h による位置の指定が外れ、所望の要素を適切に指示できなくなる場合があった。

従来、このような場合に構造化文書中の所望の要素を適切に指示し続けるためには、人手を介して X P a t h の記述を修正しなければならず、多大な手間を要し、この構造化文書を利用するシステムの開発者にとって大きな負担となっていた。

【 0 0 0 5 】

そこで、本発明は、構造化文書中の所定の要素が指示されている場合に、この構造化文書が変更されても所望の要素を適切に指示し続けることができるようにすることを目的とする。

また本発明は、X P a t h を用いて所定の要素が指示されている構造化文書が変更された場合に、この構造化文書の変更に応じて X P a t h を自動的に更新する手段を提供することを目的とする。

【 0 0 0 6 】

【課題を解決するための手段】

上記の目的を達成する本発明は、構造化文書の所定の要素を指示する、次のようなデータ処理方法として実現される。すなわち、このデータ処理方法は、所定

の指示情報により要素を指示された構造化文書が変更された場合に、この構造化文書を入力して変更内容を解析するステップと、解析された構造化文書の変更内容に応じて、変更後の構造化文書中の対応する要素を指示するように指示情報を更新するステップとを含むことを特徴とする。

【0007】

より詳細には、この構造化文書の変更内容を解析するステップは、変更前後の構造化文書をツリー構造データに変換する処理と、このツリー構造データの差分を計算する処理とを含む。そして、このツリー構造データの差分を用いて指示情報を更新する。

さらに詳しくは、このツリー構造データの差分を計算する処理では、この構造化文書の変更に応じて変形されたツリー構造データの構成要素のうちで、このツリー構造データの変形に要する操作において移動された構成要素を追跡するようにこのツリー構造データの差分を計算する。

【0008】

また好ましくは、構造化文書中の要素を指示する指示情報としてX P a t hを用いることができる。

そして、指示情報を更新する処理では、変更前後の構造化文書の差分に基づいて、指示情報が記述されたX P a t hを構成するロケーションステップを再生成することにより、X P a t hの記述を更新する。

【0009】

また、上記の目的を達成する他の本発明は、このようなデータ処理を実行する指示情報生成システムとしても実現される。すなわち、この指示情報生成システムは、複数の構造化文書の差分を計算する差分計算部と、この差分計算部により計算された差分に関する情報に基づいて所定の構造化文書における所定の部分を指示する指示情報から他の構造化文書中の対応する部分を指示する指示情報を生成する指示情報生成部とを備えることを特徴とする。

より好ましくは、この指示情報生成システムは、構造化文書の構造を解析してツリー構造データに変換する文書解析部をさらに備え、差分計算部は、この文書解析部にて変換された複数の構造化文書に対応する複数のツリー構造データを比

較して差分を計算する。

【 0 0 1 0 】

また、本発明は、このような指示情報の更新を実現するため、次のような2つのツリー構造データの差分計算方法として実現することができる。すなわち、この差分計算方法は、メモリから処理対象である2つのツリー構造データを読み出して比較し、一方のツリー構造データから他方のツリー構造データに変形するための操作をツリー構造の構成要素に対する所定の操作の組合せで表現した操作列のリストを作成するステップと、この操作列のリストのうちで、ツリー構造の構成要素の移動として把握できる操作列を移動の操作に変更するステップとを含むことを特徴とする。

ここで、ツリー構造の構成要素とは、ツリーのノード及び部分的なサブツリーを含む。また、ツリー構造の構成要素に対する所定の操作の組合せとは、例えばこれらの構成要素を追加し、削除し、または変更する操作の組合せといった、基本操作である。

【 0 0 1 1 】

また、より詳細には、このリスト中の操作列を変更するステップでは、このリスト中に、ツリー構造データにおける同一の構成要素を削除する操作と追加する操作とが存在する場合に、これらの操作に替えて構成要素を移動する操作をこのリストに加える。

さらに、このリスト中に、ツリー構造データにおける所定の構成要素を変更する操作が存在する場合に、所定の規則に基づいて、この操作を構成要素の移動を含む別の操作に置き替える。

【 0 0 1 2 】

また、上記の目的を達成するさらに他の本発明は、所定のHTML文書に対して設定されたアノテーションデータを管理する、次のようなアノテーションサーバとしても実現される。このアノテーションサーバは、アノテーションデータが設定されたHTML文書が変更された場合に、変更前後のかかるHTML文書の差分を計算する差分計算手段と、この差分計算手段の計算により得られた差分情報に基づいて、アノテーションデータをHTML文書の所定の部分に対応付ける

X P a t h を更新する X P a t h 更新手段とを備えることを特徴とする。

【 0 0 1 3 】

さらにまた、本発明は、コンピュータを制御して上記のデータ処理方法や差分計算方法における各ステップに対応する処理を実行させるプログラム、あるいは上述した指示情報更新システムやアノテーションサーバとしてコンピュータを機能させるプログラムとしても実現される。このプログラムは、磁気ディスクや光ディスク、半導体メモリ、その他の記録媒体に格納して配布したり、ネットワークを介して配信したりすることにより提供することができる。

【 0 0 1 4 】

【発明の実施の形態】

以下、添付図面に示す実施の形態に基づいて、この発明を詳細に説明する。

図 1 は、本実施の形態による X P a t h 更新方法を実現するのに好適なコンピュータ装置のハードウェア構成の例を模式的に示した図である。

図 1 に示すコンピュータ装置は、演算手段である CPU (Central Processing Unit: 中央処理装置) 1 0 1 と、M / B (マザーボード) チップセット 1 0 2 及び CPU バスを介して CPU 1 0 1 に接続されたメインメモリ 1 0 3 と、同じく M / B チップセット 1 0 2 及び A G P (Accelerated Graphics Port) を介して CPU 1 0 1 に接続されたビデオカード 1 0 4 と、P C I (Peripheral Component Interconnect) バスを介して M / B チップセット 1 0 2 に接続されたハードディスク 1 0 5、ネットワークインターフェイス 1 0 6 及び U S B ポート 1 0 7 と、さらにこの P C I バスからブリッジ回路 1 0 8 及び I S A (Industry Standard Architecture) バスなどの低速なバスを介して M / B チップセット 1 0 2 に接続されたフロッピーディスクドライブ 1 0 9 及びキーボード / マウス 1 1 0 とを備える。

なお、図 1 は本実施の形態を実現するコンピュータ装置のハードウェア構成を例示するに過ぎず、本実施の形態を適用可能であれば、他の種々の構成を取ることができる。例えば、ビデオカード 1 0 4 を設ける代わりに、ビデオメモリのみを搭載し、CPU 1 0 1 にてイメージデータを処理する構成としても良いし、A T A (AT Attachment) などのインターフェイスを介して C D - R O M (Compact

Disc Read Only Memory) や DVD-ROM (Digital Versatile Disc Read Only Memory) のドライブを設けても良い。

【 0 0 1 5 】

図 2 は、図 1 に示したコンピュータ装置にて実現される本実施の形態による X P a t h 更新システムの構成を示す図である。

図 2 に示すように、本実施の形態による X P a t h 更新システムは、構造化文書の構造を解析する文書解析部 1 0 と、文書解析部 1 0 の解析結果に基づいて構造化文書に対してなされた変更を調べる差分計算部 2 0 と、差分計算部 2 0 の計算結果に基づいて指示情報としての X P a t h の記述を更新する X P a t h 更新部 3 0 とを備えている。

【 0 0 1 6 】

上記の各構成要素は、図 1 に示したメインメモリ 1 0 3 に展開されたプログラムにて CPU 1 0 1 を制御することにより実現される仮想的なソフトウェアブロックである。CPU 1 0 1 を制御してこれらの機能を実現させる当該プログラムは、磁気ディスクや光ディスク、半導体メモリ、その他の記憶媒体に格納して配布したり、ネットワークを介して配信したりすることにより提供される。本実施の形態では、図 1 に示したネットワークインターフェイス 1 0 6 やフロッピーディスクドライブ 1 0 9、図示しない CD-ROM ドライブなどを介して当該プログラムを入力し、ハードディスク 1 0 5 に格納する。そして、ハードディスク 1 0 5 に格納されたプログラムをメインメモリ 1 0 3 に読み込んで展開し、CPU 1 0 1 にて実行することにより、図 2 に示した各構成要素の機能を実現する。

また、処理対象である構造化文書や X P a t h は、例えばハードディスク 1 0 5 の所定の領域に格納されており、本実施の形態による X P a t h の更新処理を実行するために、CPU 1 0 1 によって読み出される。

【 0 0 1 7 】

図 2 に示した本実施の形態において、文書解析部 1 0 は、構造化文書を解析し、DOM ツリーなどのツリー構造の形をしたデータ（以下、単にツリーと記す）に変換する。変換対象は、変更が行われた構造化文書における変更前後の各文書である。すなわち、文書解析部 1 0 は、ハードディスク 1 0 5 等の記憶手段に格

納されている処理対象の構造化文書（変更が行われた構造化文書）に関して、変更前の文書（以下、変更前文書）Pと変更後の文書（以下、変更後文書）P'とを読み出して解析し、それぞれツリーT（変更前文書Pに対応）とツリーT'（変更後文書P'に対応）を生成し出力する。出力されたツリーT、T'は、メインメモリ103等の記憶手段に一時的に保持され、差分計算部20にて利用される。構造化文書のツリーを生成する変換アルゴリズムとしては、既存の従来技術を利用することができる。

【0018】

差分計算部20は、文書解析部10により変換された、変更前後の各構造化文書（ツリー）の差分を計算する。これにより、処理対象である構造化文書に対してなされた変更の内容が認識される。本実施の形態では、差分の計算方法として、後に行うXPathの更新に適した新規な方法を提案する。以下、この差分計算アルゴリズムについて説明する。

まず、前提として、従来の一般的な差分計算のアルゴリズムを説明する。2つのツリーの差分を計算するアルゴリズムとしては、従来から種々のアルゴリズムが提案されているが、典型的な差分計算アルゴリズムに、最小コスト操作列を算出するものがある。

【0019】

図16、17は、ツリーの差分計算の例を示す図である。

図16に示すように、ノードaを親とし、ノードb、c、dを子とするツリー構造で、ノードb、dの位置が入れ替わった2つのツリー161、162の差分を計算する場合を考える。なお、ツリーを変形する操作における処理コストは、基本操作である所定のノードを削除するRemoveNode、所定のノードを追加するInsertNode、ノードの内容を変更するModifyにおいて、それぞれ1とする。

このとき、最小コスト操作列を算出するアルゴリズムでは、ツリー161からツリー162へ変形するためには、ノードbの内容をノードdの内容に変更し（Modify(b→d)）、ノードdの内容をノードbの内容に変更する（Modify(d→b)）という操作が行われると計算する。これは、上記の処理コストの値から、処理コスト2でツリー161からツリー162へ変形することができ、これが最小の値

だからである。

【 0 0 2 0 】

また、図 1 7 に示すように、ノード a を親とし、ノード b、c、d を子とするツリー構造で、ノード c、d に対するノード b の位置が異なる 2 つのツリー 1 7 1、1 7 2 の差分を計算する場合を考える。

このとき、最小コスト操作列を算出するアルゴリズムでは、ツリー 1 7 1 からツリー 1 7 2 へ変形するためには、ツリー 1 7 1 からノード b を削除し (Remove Node(b))、ツリー 1 7 2 における図示の位置にノード b を追加する (InsertNode(b)) という操作が行われると計算する。これも、上記の処理コストの値から、最小の処理コスト 2 でツリー 1 7 1 からツリー 1 7 2 へ変形することができるためである。

【 0 0 2 1 】

これに対し、本実施の形態では、この差分計算が X P a t h の自動更新のために行われることから、かかる最小コスト操作列を算出するアルゴリズムは好適な手法とは言えない。図 1 6 の例では、ツリー 1 6 1 に対して X P a t h がノード b を指していたと仮定すると、ノードの内容が (b から d へ) 変更されたという情報のみでは、X P a t h の指示先を適切に変更することはできない。この例の場合、ツリー 1 6 1 からツリー 1 6 2 への変換は、ノード b とノード d とを入れ替えるように移動する操作が行われたと見ることができ、この情報に基づけば、ノード b を指す X P a t h の記述を適切に変更することが可能である。したがって、X P a t h を更新するための情報として用いるには、ツリー 1 6 1 からツリー 1 6 2 への変換を表す操作列は、ノードを移動する操作の MoveNode を用いて MoveNode(b)、MoveNode(d) と表す方が適している。しかしながら、MoveNode は、上述した基本操作 RemoveNode、InsertNode の組合せ (ノードを削除し、追加する操作) であるため、ツリー 1 6 1 からツリー 1 6 2 への変換に要するトータルコストは 4 となる。したがって、最小コスト操作列を算出するアルゴリズムでは MoveNode を検出することはできない。

【 0 0 2 2 】

同様に、図 1 7 の例においても、ノード b を削除する操作またはノード b を追

加する操作の一方の情報から X P a t h の指示先を適切に変更することはできない。そして、ツリー 1 7 1 からツリー 1 7 2 への変換は、ノード b をノード c の左からノード d の右へ移動する操作が行われたと見ることができ、この情報に基づけば、ノード b を指す X P a t h の記述を適切に変更することが可能である。したがって、ツリー 1 7 1 からツリー 1 7 2 への変換を表す操作列は、MoveNode (b) と表す方が適している。この例では、MoveNode は、RemoveNode、InsertNode の組合せであることからトータルコストは 2 で変わらない。しかし、最小コスト操作列を算出するアルゴリズムで MoveNode を検出できる保証はなく、やはり本実施の形態には適さない。

なお、以上の考察は、サブツリー（ツリー中の部分的なツリー構造）を移動する操作の MoveTree についても同様である。以下では簡単のため、個々のノードに対する処理についてのみ説明するが、同様の解析を MoveTree に対しても行うことができるのは言うまでもない。

【 0 0 2 3 】

以上のことを踏まえて、差分計算部 2 0 にて実行される、本実施の形態に好適な差分計算アルゴリズムを説明する。本実施の形態で用いられる差分計算アルゴリズムは、ツリーの変形において移動されたもの（ノードやサブツリー）を追跡することを方針とする。

図 3 乃至図 6 は、本実施の形態に好適な差分計算アルゴリズムによる処理を説明するフローチャートである。

差分計算部 2 0 は、メインメモリ 1 0 3 等の記憶手段に一時的に保持されている、変更前文書 P に対応するツリー T と変更後文書 P' に対応するツリー T' とを入力し、基本操作である 3 つの操作 RemoveNode、InsertNode 及び Modify を用いて、ツリー T からツリー T' へ変形するために必要な操作を解析し、得られた操作列のリスト L を生成する。この解析には、上述した最小コスト操作列を算出するアルゴリズムなど、既存の手法を用いることができる。生成された操作列のリスト L は、メインメモリ 1 0 3 等の記憶手段に一時的に保持される。そして次に、差分計算部 2 0 は、メインメモリ 1 0 3 等に保持された操作列のリスト L に対して図 3 乃至図 6 に示す解析を行って、MoveNode を検出する。

【 0 0 2 4 】

図 3 に示す InsertNode 解析において、差分計算部 2 0 は、まずリスト L 中の所定の InsertNode (InsertNode(n)) に着目する (ステップ 3 0 1)。そして、着目した InsertNode の対象ノード (図示の例ではノード n) と同一のノードに対する操作 RemoveNode がリスト L 中に存在するかどうかを調べる (ステップ 3 0 2)。このとき、対応する RemoveNode (すなわち、RemoveNode(n)) が存在しないならば、ノード n は、ツリー T' において新たに追加されたノードであるので、そのまま処理を終了する。

一方、RemoveNode(n) が存在する場合は、これと InsertNode(n) との組合せにより MoveNode(n) が構成されるので、リスト L に MoveNode(n) を追加し (ステップ 3 0 3)、リスト L から InsertNode(n) と RemoveNode(n) とを削除する (ステップ 3 0 4)。差分計算部 2 0 は、以上の処理をリスト L 中の全ての InsertNode に対して行う。

【 0 0 2 5 】

図 4 に示す RemoveNode 解析において、差分計算部 2 0 は、まずリスト L 中の所定の RemoveNode (RemoveNode(n)) に着目する (ステップ 4 0 1)。そして、着目した RemoveNode の対象ノード (図示の例ではノード n) と同一のノードに対する操作 InsertNode がリスト L 中に存在するかどうかを調べる (ステップ 4 0 2)。このとき、対応する InsertNode (すなわち、InsertNode(n)) が存在しないならば、ノード n は、ツリー T' において削除されたノードであるので、そのまま処理を終了する。

一方、InsertNode(n) が存在する場合は、これと RemoveNode(n) との組合せにより MoveNode(n) が構成されるので、リスト L に MoveNode(n) を追加し (ステップ 4 0 3)、リスト L から RemoveNode(n) と InsertNode(n) とを削除する (ステップ 4 0 4)。差分計算部 2 0 は、以上の処理をリスト L 中の全ての RemoveNode に対して行う。

【 0 0 2 6 】

図 5 及び図 6 に示す Modify 解析において、差分計算部 2 0 は、まずリスト L 中のノード n 1 の内容を変更する操作 Modify(n1→nx) に着目する (ステップ 5 0 1

）。そして、反対にノードの内容を n_1 にする変更を行う操作 $\text{Modify}(n_y \rightarrow n_1)$ がリスト L 中に存在するかどうかを調べる（ステップ 5 0 2）。

$\text{Modify}(n_y \rightarrow n_1)$ が存在するならば、次にノード n_x の内容とノード n_y の内容とが同一（すなわち、 $n_x = n_y$ ）かどうかを調べる（ステップ 5 0 3）。 $n_x = n_y$ である場合、ノード n_1 とノード $n_x (= n_y)$ の位置が入れ替わったと解釈できるので、リスト L に $\text{Movenode}(n_1)$ 、 $\text{Movenode}(n_y)$ を追加し（ステップ 5 0 4）、リスト L から $\text{Modify}(n_1 \rightarrow n_x)$ と $\text{Modify}(n_y \rightarrow n_1)$ とを削除する（ステップ 5 1 3）。

一方、 $n_x \neq n_y$ でない場合、ノード n_1 がツリー T におけるノード n_y のあった位置に移動し、当該ノード n_y は削除され、元のノード n_1 の位置に他のノード n_x が新たに追加されたと解釈できる。そこで、リスト L に $\text{InsertNode}(n_x)$ 、 $\text{RemoveNode}(n_y)$ 、 $\text{Movenode}(n_1)$ を追加し（ステップ 5 0 5）、さらに図 3、4 に示した InsertNode 解析、 RemoveNode 解析を実行する（ステップ 5 0 6）。この後、リスト L から $\text{Modify}(n_1 \rightarrow n_x)$ と $\text{Modify}(n_y \rightarrow n_1)$ とを削除する（ステップ 5 1 3）。

【 0 0 2 7 】

また、ステップ 5 0 2 で $\text{Modify}(n_y \rightarrow n_1)$ が存在しない場合、次に差分計算部 2 0 は、ノード n_1 に対する操作 $\text{InsertNode}(n_1)$ がリスト L 中に存在するかどうかを調べる（図 6、ステップ 5 0 7）。存在するならば、ノード n_1 が他の所定の位置に移動し、元のノード n_1 の位置に他のノード n_x が新たに追加されたと解釈できる。そこで、リスト L に $\text{InsertNode}(n_x)$ 、 $\text{Movenode}(n_1)$ を追加し（ステップ 5 0 8）、さらに図 3 に示した InsertNode 解析、を実行する（ステップ 5 0 9）。この後、リスト L から $\text{Modify}(n_1 \rightarrow n_x)$ と $\text{InsertNode}(n_1)$ とを削除する（図 5、ステップ 5 1 3）。

【 0 0 2 8 】

さらに、ステップ 5 0 7 で $\text{InsertNode}(n_1)$ がリスト L 中に存在しない場合、次に差分計算部 2 0 は、ノード n_1 に対する操作 $\text{RemoveNode}(n_x)$ がリスト L 中に存在するかどうかを調べる（図 6、ステップ 5 1 0）。存在するならば、ノード n_1 が削除され、同じ位置にノード n_x が移動されたと解釈できる。そこで、リス

トLにRemoveNode(n1)、Movenode(nx)を追加し(ステップ511)、さらに図3に示したRemoveNode解析、を実行する(ステップ512)。この後、リストLからModify(n1→nx)とRemoveNode(nx)とを削除する(図5、ステップ513)。

【0029】

ステップ510においてRemoveNode(nx)がリストL中に存在しない場合、単にノードn1の内容がnxに変更されたと解釈できるので、そのまま処理を終了する。差分計算部20は、以上の処理をリストL中の全てのModifyに対して行う。

以上のようにして、ツリーT、T'の差分が計算される。得られた差分データは、メインメモリ103等の記憶手段に一時的に保持され、XPath更新部30にて利用される。上記の3種類の解析にて実現されるように、本実施の形態では、ツリーTからツリーT'への変形において、ノードの移動として解釈できる操作はできるだけ移動操作moveとして検出することにより、後のXPathの更新処理に利用できるようにする。

【0030】

XPath更新部30は、差分計算部20によるツリーTとツリーT'との差分の計算結果と、変更前文書Pに対応するXPath(以下、XPath(P)と記す)とを入力し、これらに基づいて変更後文書P'に対応するXPath(以下、XPath(P')と記す)を生成し出力する。

図7は、XPath更新部30の機能構成を説明する図である。

図7を参照すると、指示情報生成手段であるXPath更新部30は、XPathを解釈する機能(XPath解釈器31)と、変更前後の構造化文書におけるノードの対応に関する情報を保持する機能(ノード対応表32)と、XPathを生成する機能(XPath生成器33)とを備える。そして、メインメモリ103やハードディスク105等の記憶手段から処理対象である更新前文書P、更新後文書P'、更新前文書Pと更新後文書P'との間の差分D、XPath(P)を入力し、上記の各機能を用いてXPath(P')を生成する。生成されたXPath(P')は、ハードディスク105等の記憶手段に格納される。

【0031】

以下、XPath更新部30によるXPathの更新処理を詳細に説明する。

図8は、更新前文書Pと更新後文書P' との間のノード対応関係を示す図である。

X P a t h (P) は、各パスの階層（ロケーションステップ：LocationStep） $L s(i)$ ($i = 0, 1, 2, \dots, n$) によって構成される。更新前文書Pにおいて、ロケーションステップ $L s(i)$ の指示対象となるノードの集合（ノードセット：NodeSet） $S(i)$ が、X P a t h 解釈器31の処理によって計算される。同様に、更新後文書P' において、ロケーションステップ $L s(i)$ の指示対象となるノードセット $S(i)'$ が計算される。

一方、更新前文書P、更新後文書P' 、更新前文書Pと更新後文書P' との間の差分Dから、更新前文書Pと更新後文書P' との間のノード対応関係を表すノード対応表32が作成される。作成されたノード対応表32は、図1に示したコンピュータ装置におけるCPUの101のレジスタやメインメモリ103等に保持される。ノード対応表32の例を図10に示す。図示のノード対応表32によれば、例えば、更新前文書PのノードN0が更新後文書P' のノードN' 0に対応しており、更新前文書PのノードN0は更新後文書P' に対応するノードがない（すなわち構造化文書の変更によって削除された）ことがわかる。

【0032】

このノード対応表32と、更新前文書Pにおけるロケーションステップ $L s(i)$ の指示対象であるノードセット $S(i)$ とに基づいて、ノードセット $S(i)''$ が求まる。

図9は、ノードセット $S(i)$ とノードセット $S(i)''$ との対応関係を示す図である。

ノードセット $S(i)'$ とノードセット $S(i)''$ との違いは、ノードセット $S(i)'$ が単純にパスのパターンを更新後文書P' に適用させて求めたノードセットであるのに対し、ノードセット $S(i)''$ は、差分情報により変更を追跡して求めたノードセットであるということである。また、ノードセット $S(i)'$ 、 $S(i)''$ とともに変更後文書P' におけるノードセットである。

【0033】

次に、X P a t h 生成器33により、このノードセット $S(i)'$ 、 $S(i)''$ を比

較して、 $XP ath(P)$ におけるロケーションステップ $Ls(i)$ を更新する。更新処理の詳細は後述する。この手順を $i=0 \sim n$ まで繰り返すことによりロケーションステップ $Ls(j)'$ ($j=0, 1, 2, \dots, m$)を得る。このロケーションステップ $Ls(j)'$ は直接、更新された $XP ath(P')$ を表す。

図11は、 $XP ath$ 生成器33による $XP ath$ （ロケーションステップ）の生成手順を示すフローチャートである。

図11を参照すると、 $XP ath$ 生成器33は、まず、ノードセット $S(i)'$ 、 $S(i)''$ を比較する（ステップ1101）。そして、ノードセット $S(i)'$ とノードセット $S(i)''$ とが等しいか、またはノードセット $S(i)'$ がノードセット $S(i)''$ に含まれるのであれば、ロケーションステップ $Ls(i)$ は変更する必要はなく、そのままロケーションステップ $Ls(j)'$ として出力する（ステップ1102、1103）。

一方、ノードセット $S(i)''$ がノードセット $S(i)'$ に含まれる場合、 $XP ath$ 生成器33は、次にロケーションステップ $Ls(j-1)'$ で示されるノードから、ノードセット $S(i)''$ に含まれる各ノードへのロケーションステップを生成する（ステップ1103、1104）。

以上のようにして、変更後文書 P' に対応したロケーションステップが生成され、 $XP ath(P)$ が $XP ath(P')$ へ更新される。

【0034】

ところで、 $XP ath$ の表記方法によっては、ステップ1104で生成されたロケーションステップを、所定の一般化ルールに基づいて一般化することにより、簡明な表現に集約することができる。もし、用意された一般化ルールによってはロケーションステップ $Ls(j)'$ が生成できない場合は、警告を知らせるウィンドウや修正を促すウィンドウを表示するなど、エラー処理を行うと共に、ステップ1104で生成されたロケーションステップをそのまま出力することができる。

なお、ステップ1104におけるロケーションステップの生成は、例えば、下記文献1に開示された既知の戦略によって行うことができる。また、上記のロケーションステップの集約は、例えば、下記文献2に開示された既知の戦略によっ

て行うことができる。

文献 1 : 2001/11/08: A Visual Approach to Authoring XPath Expressions Accepted for Markup Languages: Theory and Practice, Vol.3, No.2 (to appear) . This is a paper originally published in the Proceedings Extreme Markup Languages 2001, pp.1-15, Montreal, Canada (14-17, August 2001).

<http://ares.trl.ibm.com/freedom/doc/extml2001/abe0114.html>

文献 2 : 2001/07/13: XSLT Stylesheet Generation by Example with WYSIWYG Editing Accepted for the presentation at International Symposium on Applications and the Internet (SAINT 2002)

<http://ares.trl.ibm.com/freedom/doc/saint2002/saint2002.html>

【 0 0 3 5 】

次に、具体的なツリーの変形例に基づいて XPath の更新方法を説明する。

図 1 2 乃至図 1 4 は、更新前文書 P と更新後文書 P' の例を示す図である。なお、図には当該構造化文書 P、P' のツリー構造が記載されている。

図 1 2 を参照すると、変更前文書 P は、ルートノード a が 3 つの子ノード b を持ち、そのうちの一番左のノード b が 2 つの子ノード c を持ち、そのうちの右側のノード c が子ノード b を持つツリー構造となっている。これに対し、変更後文書 P' は、ノード c の下に位置するノード b がノード a の子に移動した構造となっている。

【 0 0 3 6 】

ここで、変更前文書 P に対して、ノード a の子の 2 つのノード b を指す XPath (P) 「/a/b」が存在するものとする。「/a/b」という表現は、ノード a の子にある全てのノード b を指す。変更前文書 P が変更後文書 P' に変更された場合、この「/a/b」をそのまま用いると、変更後文書 P' においてノード a の子にある 4 つのノード b を指すこととなる。しかし、変更によってノード a の子に移動したノード b は、変更前文書 P においても存在し、かつこの時点では「/a/b」によっては指されていないノードであったので、変更後文書 P' においても「/a

/b」によって指されるべきではない。

【0037】

本実施の形態においてX P a t h更新部30は、差分計算部20により計算された差分Dに基づいて作成されたノード対応表32により、図13に示すように、変更前文書PにおいてX P a t h (P)に指されていた3つのノードbが変更後文書P'における4つのノードbのうち左から数えて1番目から3番目までのノードbであることがわかる。そこで、X P a t h生成器33により、これらのノードb（すなわち変更前文書Pにおいても同じ位置に存在したノードb）のみを指すように、X P a t h (P')を生成する。すなわち、「/a/b」という表現を「/a/b[position() >= 3]」という表現に更新する。

【0038】

また、図14を参照すると、図12と同様の変更前文書Pに対し、変更後文書P'は、ノードaの子の3つのノードbのうち、1つが削除されている。この場合、変更後文書P'において「/a/b」という表現で指し示されるノードの集合は、変更前文書Pにおいてかかる表現で指し示されるノードの集合に含まれる。すなわち、余分なノードを指してはいない。したがって、X P a t hの表現は変更する必要はない。なお、本実施の形態を用いた実施態様によっては、変更後文書P'において、指示するノードbの1つが削除されたことをユーザに通知するようにしても良い。

【0039】

以上説明したように、本実施の形態は、構造化文書が変更された場合に、変更前後の構造化文書の差分を検出し、これに基づいて対応するX P a t hを自動的に更新することを実現する。しかしながら、実際には、構造化文書やX P a t hを用いたシステムの開発者等の意図通りにX P a t hが更新されとは限らない。また、X P a t hが自動的に更新された後、さらに修正を施したいと考える場合もあり得る。そこで、対話的に処理を進めるX P a t h更新ツールとして実施することも可能である。

図15は、このようなX P a t h更新ツールを備えたアノテーションシステムの例を示す図である。

図15において、アノテーションサーバ1500は、本実施の形態による文書解析部10、差分計算部20及びXPath更新部30に相当する機能を備えるものとする。これらの機能は、アノテーションサーバ1500を構成するコンピュータシステムにおいて、プログラム制御されたCPUの機能として実現される。アノテーション開発者が操作するコンソール1510のディスプレイ装置には、処理対象である構造化文書（例えばHTML文書）1511とXPathを更新するための対話用ウィンドウ1512とが表示されている。

【0040】

アノテーションサーバ1500は、自身の管理下でアノテーションを付した所定の構造化文書1511が変更されると、コンソール1510のディスプレイに変更前後の構造化文書1511を表示すると共に、対話用ウィンドウ1512を表示して、この構造化文書1511の変更に応じてXPathを更新するかどうかをアノテーション開発者に問い合わせる。そして、対話用ウィンドウ1512の「Yes」ボタンがクリックされたならば、アノテーションサーバ1500の文書解析部10、差分計算部20及びXPath更新部30に相当する機能により、XPathが自動的に更新される。また、「Delete」ボタンがクリックされたならば、当該XPathが削除され、構造化文書1511のアノテーションが消去されることとなる。さらに、構造化文書1511において単に削除されたり変更されたりしたエレメント（ノード）に対するXPathは、参照不可能となる。この場合、アノテーションを付けたエレメントが削除されたことをアノテーション開発者に通知し、当該アノテーションを他のエレメントに付するかどうかを訪ねるメッセージを出力させるようにしても良い。

【0041】

なお、以上の説明においては、XPathを用いてXMLやHTMLなどの構造化文書の要素を指示する場合について説明したが、本実施の形態は、XPath以外の任意の手段を用いて構造化文書の要素を指示する場合にも応用することが可能である。すなわち、構造化文書が変更された場合、本実施の形態に示した差分計算部20に相当する機能にて当該構造化文書の変更前後の差分を取り、当該構造化文書の要素を指示する手段（指示情報等）に応じた変形を行うことによ

り、かかる構造化文書の変更に対応してその要素の指示内容を適切に更新することが可能である。

【0042】

【発明の効果】

以上説明したように、本発明によれば、構造化文書中の所定の要素が指示されている場合に、この構造化文書が変更されても所望の要素を適切に指示し続けることができる。

また、本発明によれば、X P a t hを用いて所定の要素が指示されている構造化文書が変更された場合に、この構造化文書の変更に応じてX P a t hを自動的に更新する手段を提供することができる。

【図面の簡単な説明】

【図1】 本実施の形態によるX P a t h更新方法を実現するのに好適なコンピュータ装置のハードウェア構成の例を模式的に示した図である。

【図2】 図1に示したコンピュータ装置にて実現される本実施の形態によるX P a t h更新システムの構成を示す図である。

【図3】 本実施の形態に好適な差分計算アルゴリズムによる処理を説明するフローチャートであり、InsertNode解析を説明する図である。

【図4】 本実施の形態に好適な差分計算アルゴリズムによる処理を説明するフローチャートであり、RemoveNode解析を説明する図である。

【図5】 本実施の形態に好適な差分計算アルゴリズムによる処理を説明するフローチャートであり、Modify解析を説明する図である。

【図6】 本実施の形態に好適な差分計算アルゴリズムによる処理を説明するフローチャートであり、Modify解析を説明する図である。

【図7】 本実施の形態におけるX P a t h更新部の機能構成を説明する図である。

【図8】 本実施の形態における更新前文書Pと更新後文書P' との間のノード対応関係を示す図である。

【図9】 本実施の形態におけるノードセットS(i)とノードセットS(i)"との対応関係を示す図である。

【図 1 0】 本実施の形態にて用いられるノード対応表の例を示す図である。

【図 1 1】 本実施の形態の X P a t h 生成器による X P a t h の生成手順を示すフローチャートである。

【図 1 2】 更新前文書 P と更新後文書 P' の例を示す図である。

【図 1 3】 更新後文書 P' に対応させて X P a t h を更新した様子を示す図である。

【図 1 4】 更新前文書 P と更新後文書 P' の他の例を示す図である。

【図 1 5】 本実施の形態を用いた X P a t h 更新ツールを備えたアプリケーションシステムの例を示す図である。

【図 1 6】 一般的なツリーの差分計算の例を示す図である。

【図 1 7】 一般的なツリーの差分計算の他の例を示す図である。

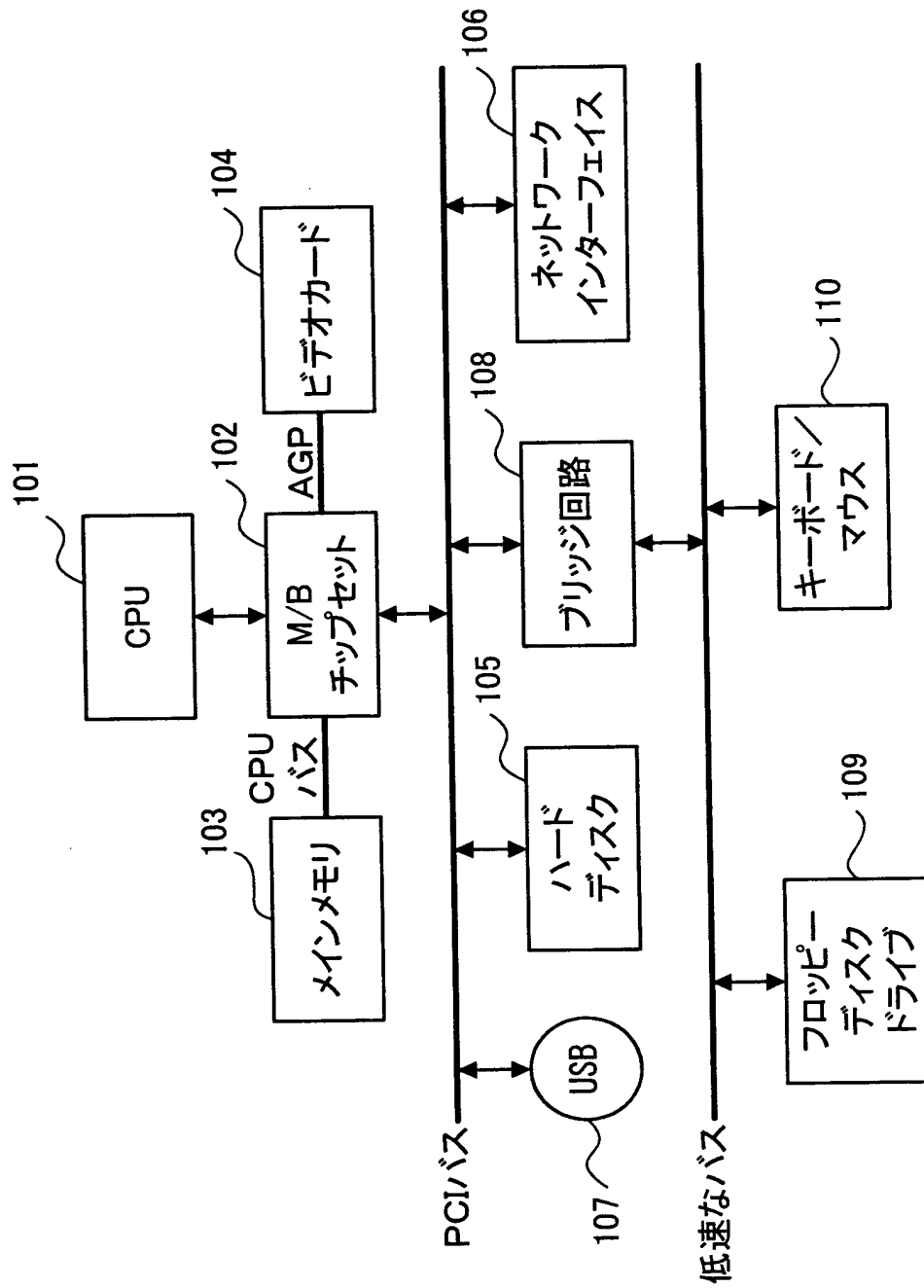
【図 1 8】 XML 文書の例を示す図である。

【符号の説明】

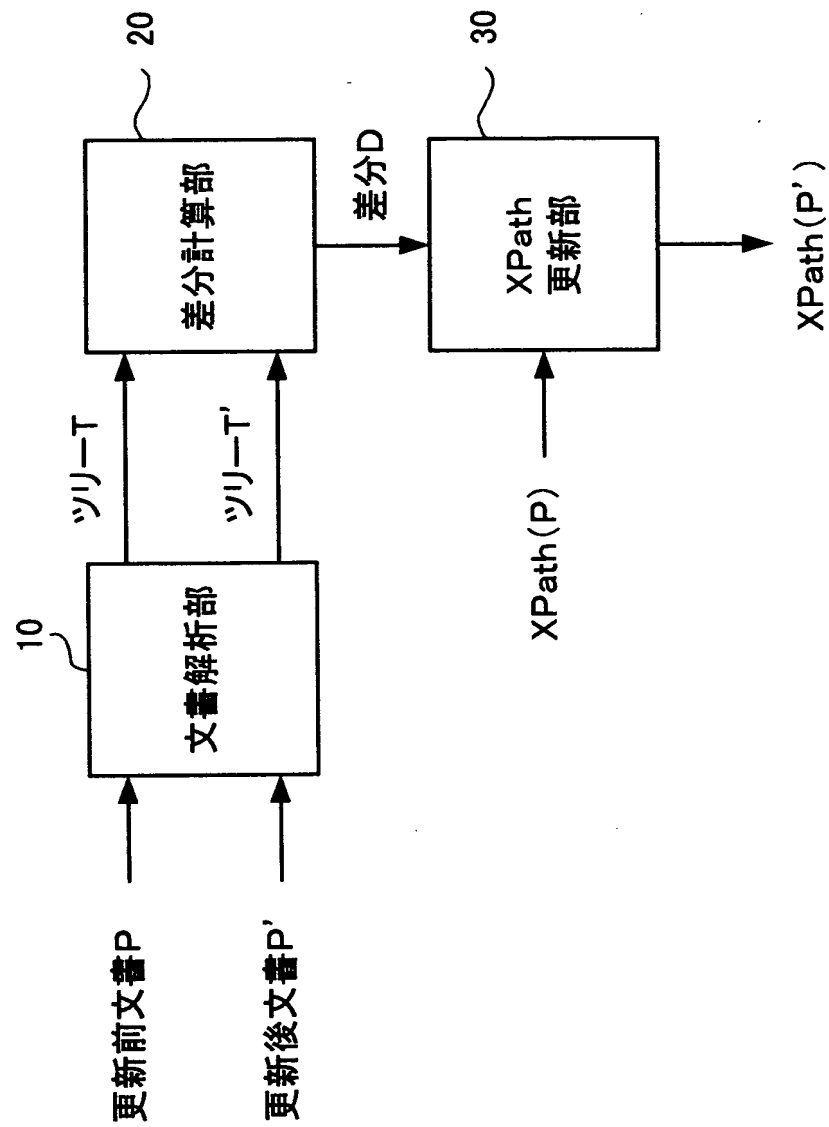
1 0 … 文書解析部、 2 0 … 差分計算部、 3 0 … X P a t h 更新部、 3 1 … X P a t h 解釈器、 3 2 … ノード対応表、 3 3 … X P a t h 生成器、 1 0 1 … C P U、 1 0 2 … M / B チップセット、 1 0 3 … メインメモリ、 1 0 5 … ハードディスク

【書類名】 図面

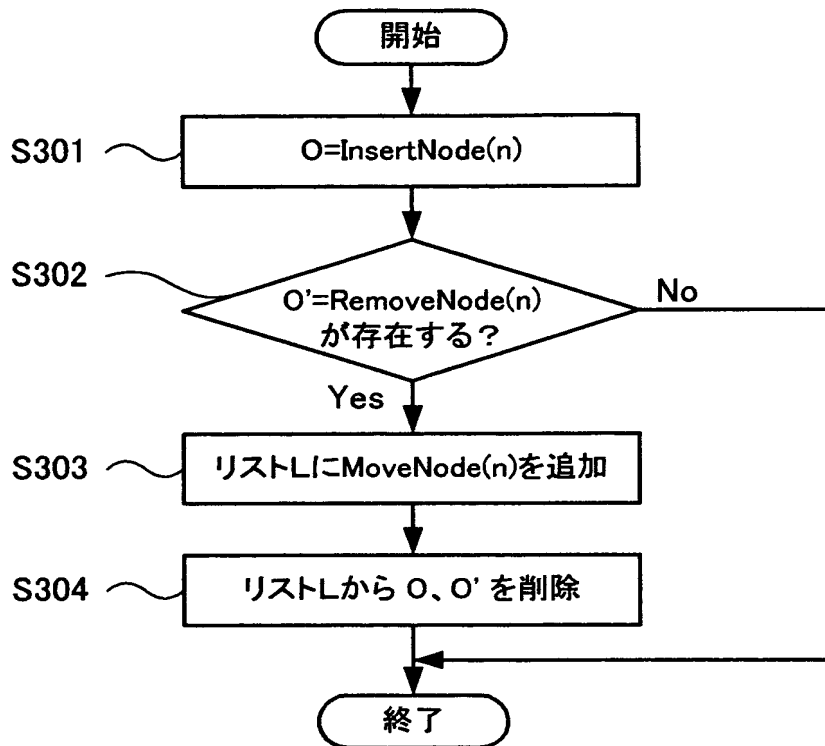
【図 1】



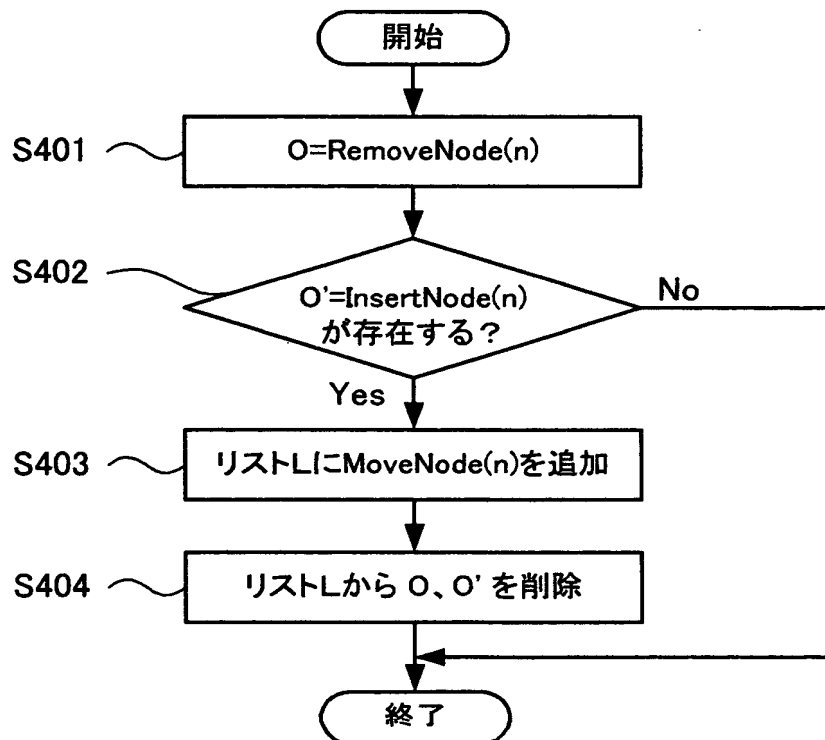
【図 2】



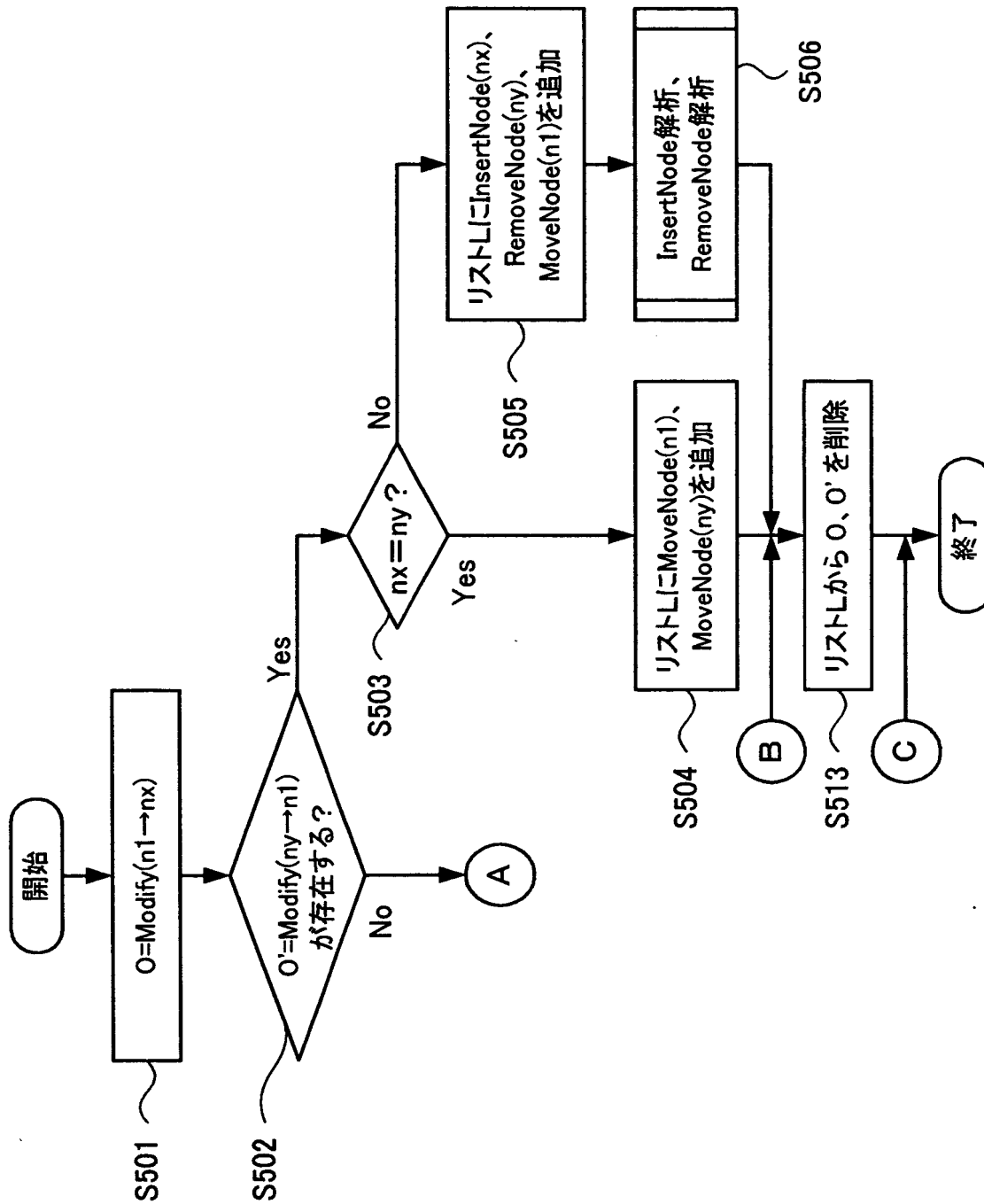
【図 3】



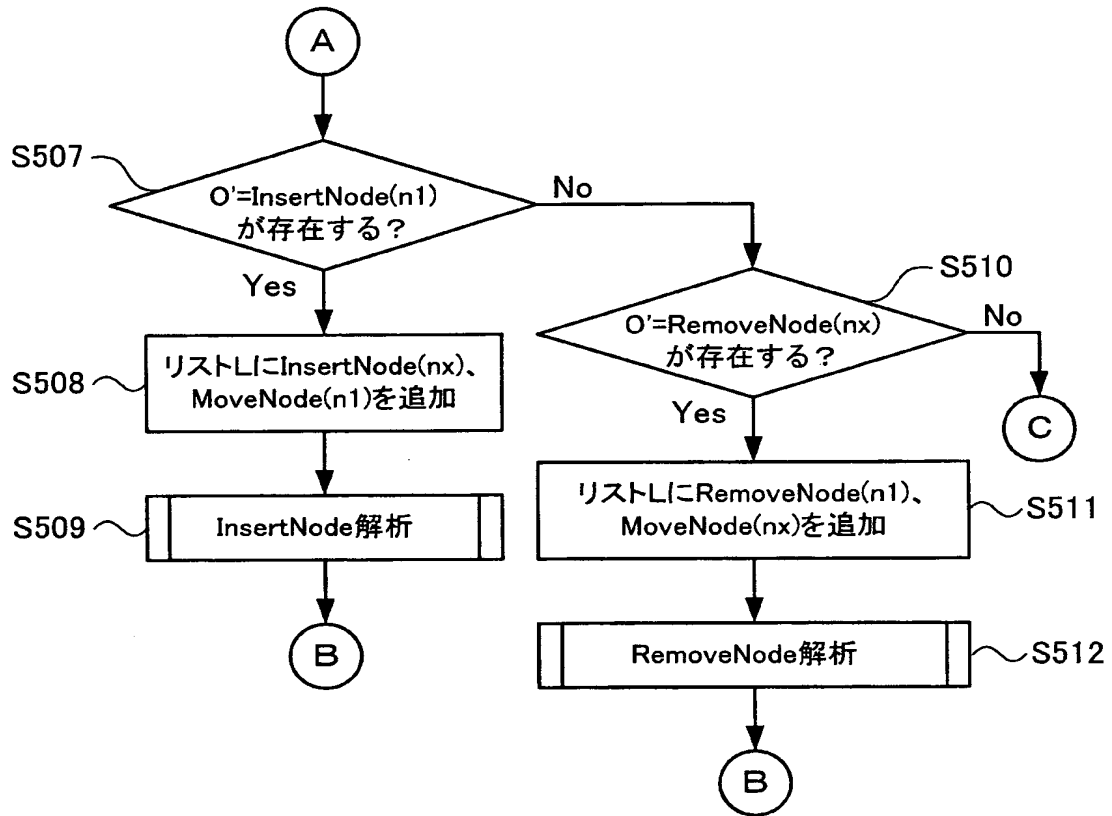
【図 4】



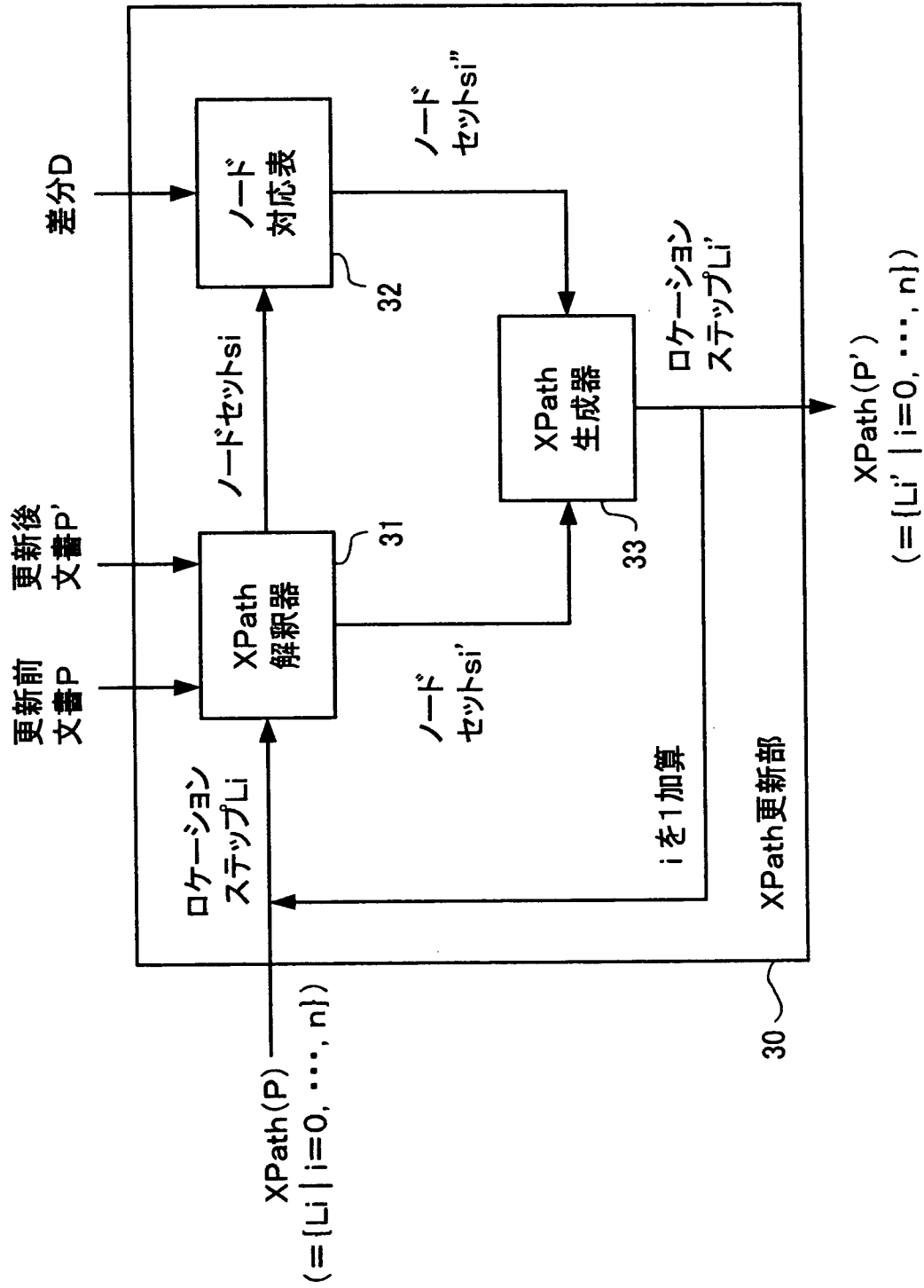
【図 5】



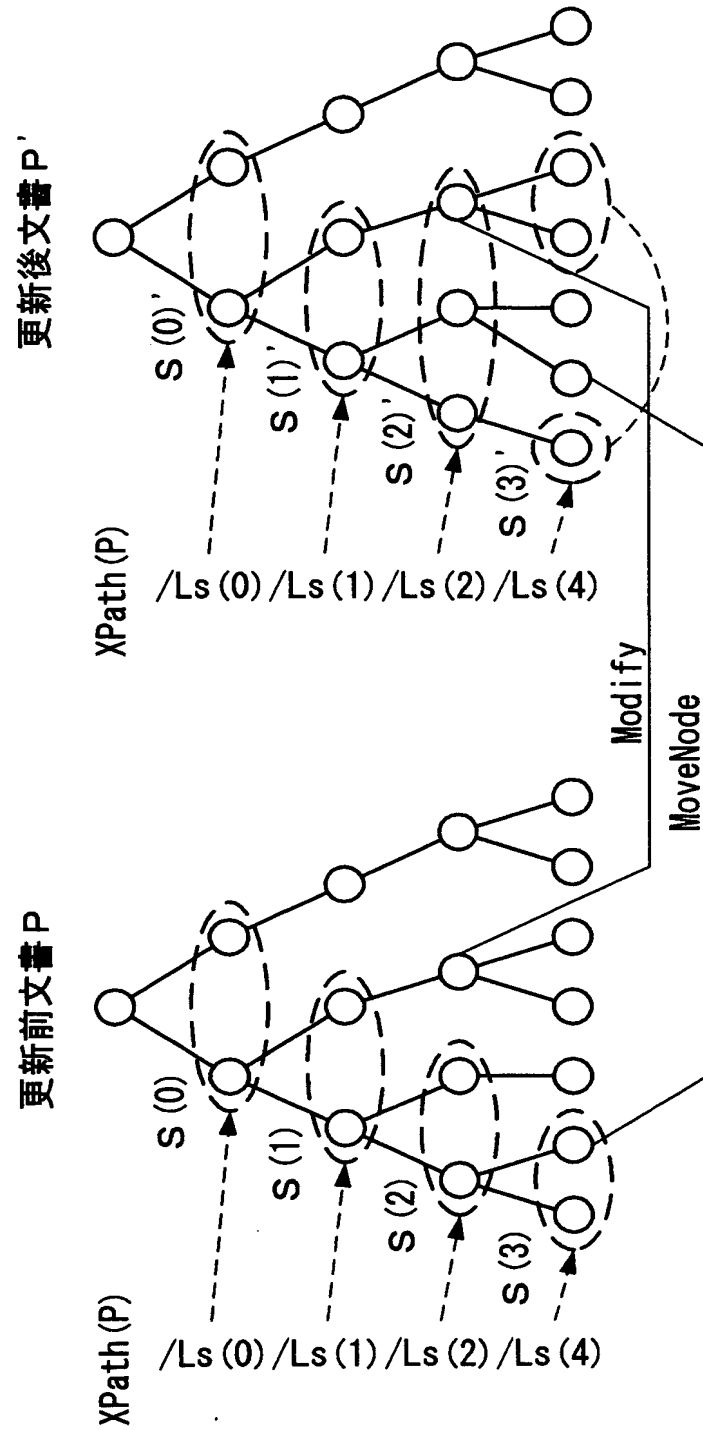
【図 6】



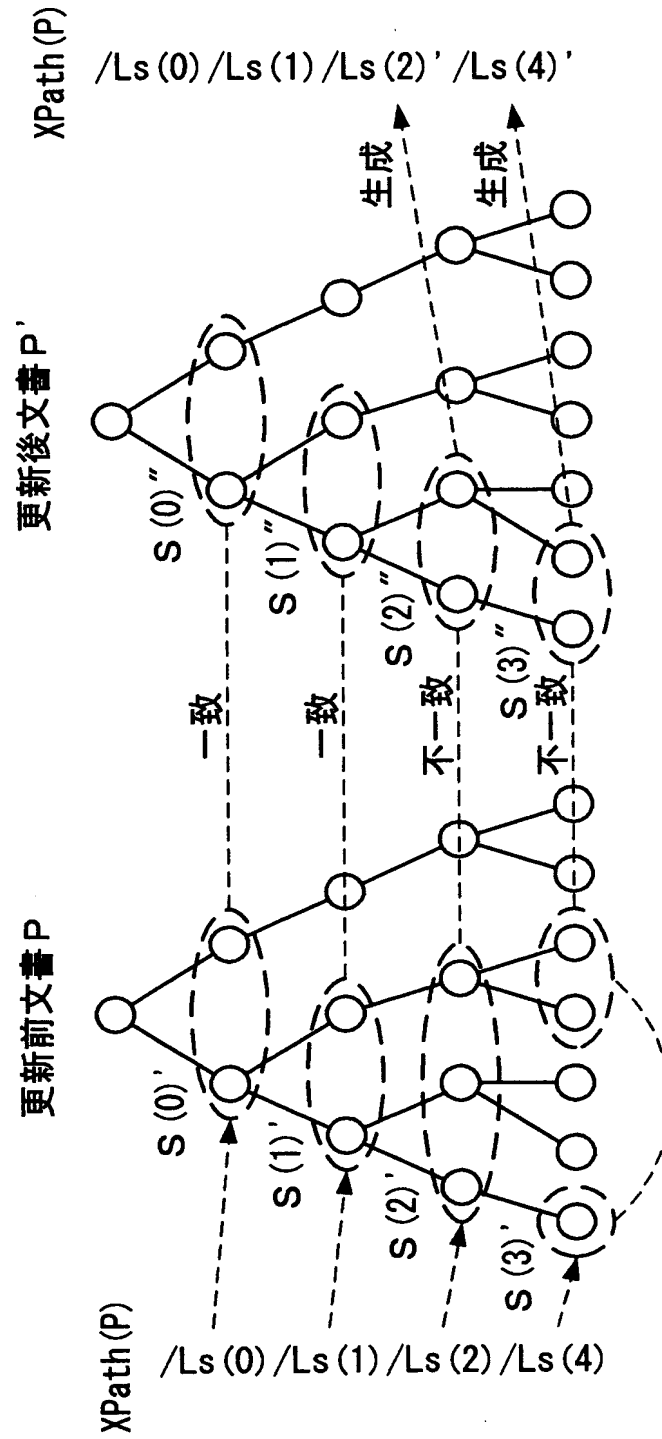
【図 7】



【図 8】



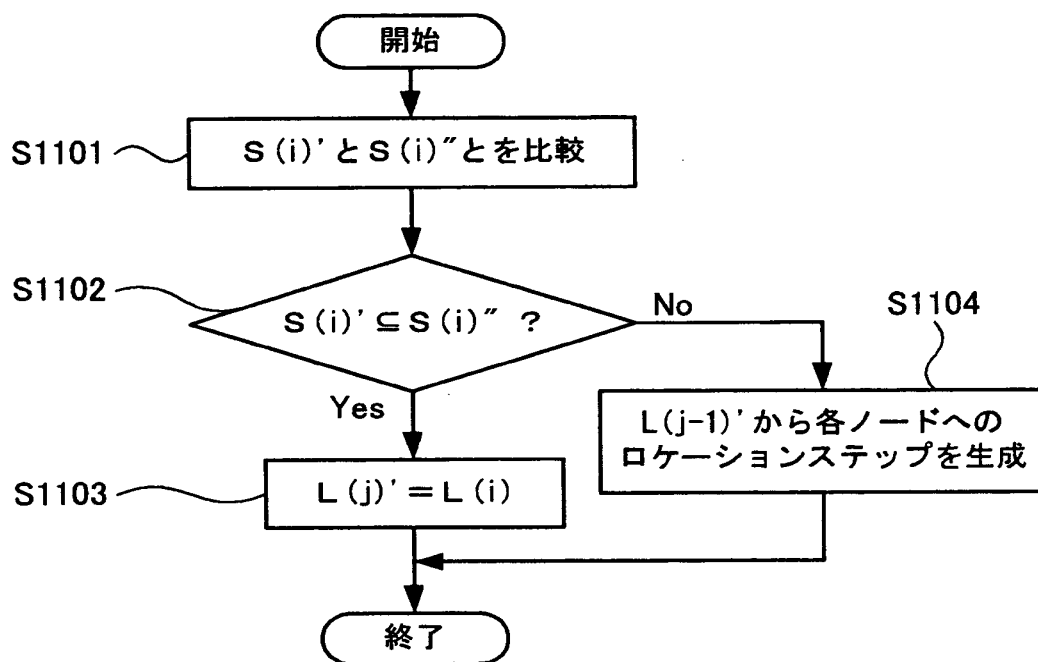
【图 9】



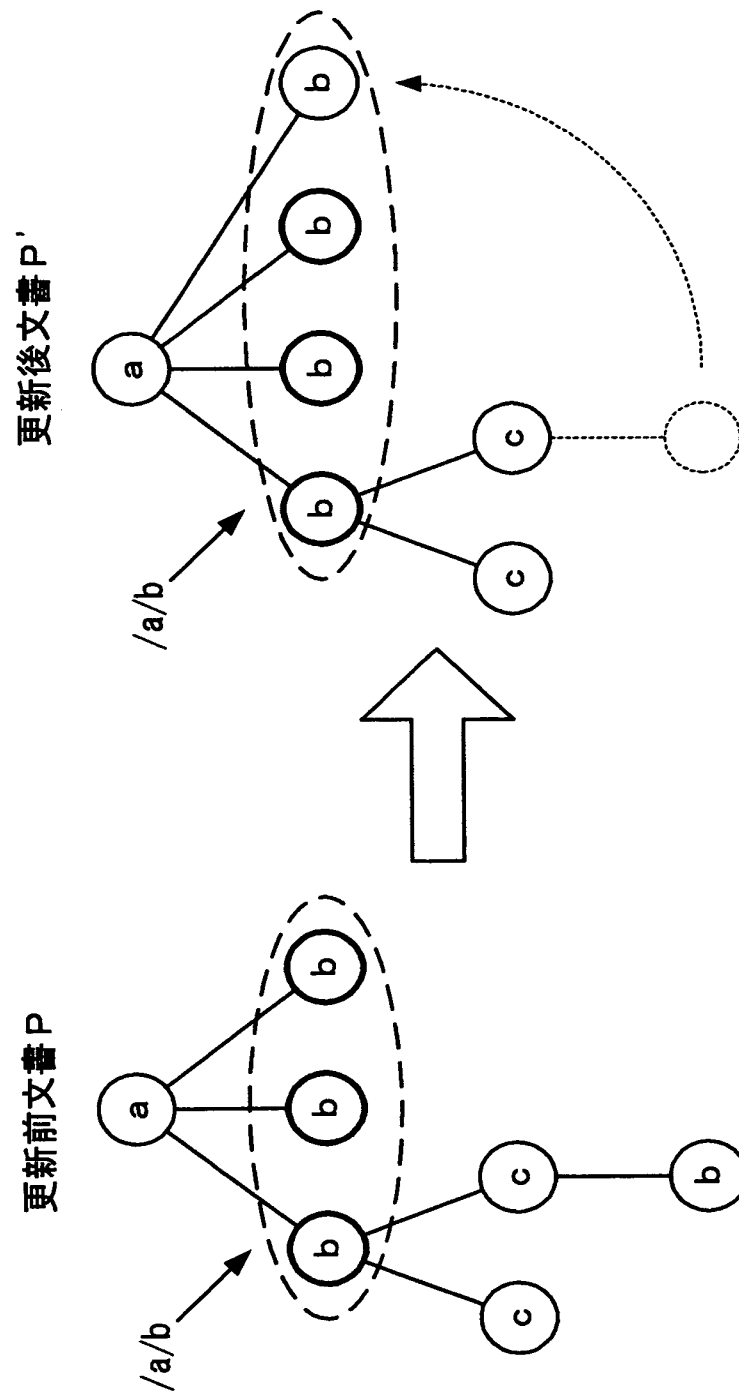
【図 10】

更新前文書のノード	更新後文書のノード
N 0	N' 0
N 1	N' 1
N 2	N' 2
N 3	(non existent)
⋮	⋮
⋮	⋮
N n	N' n

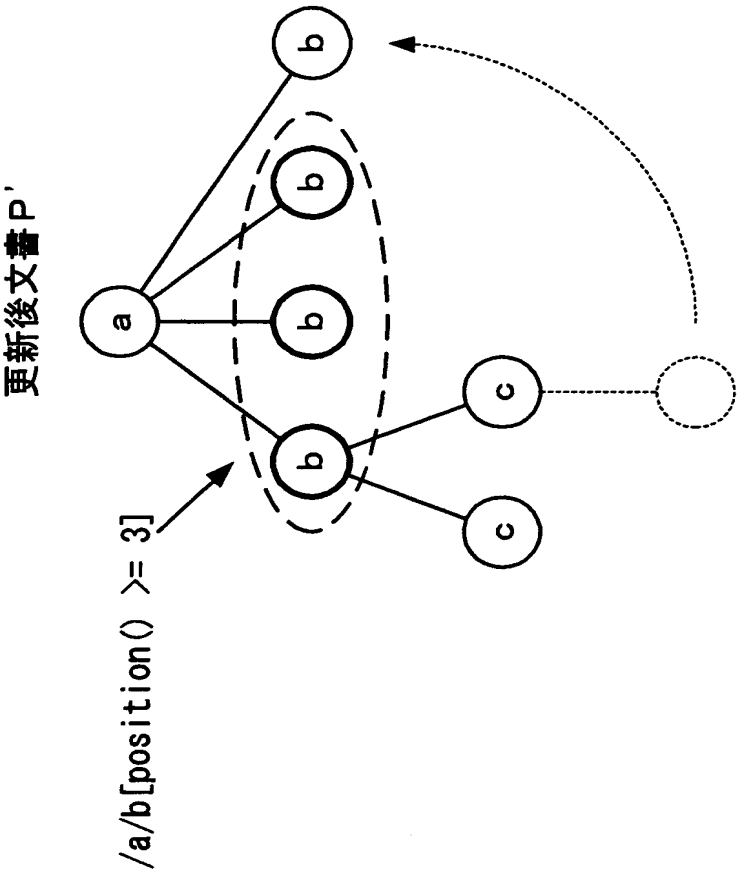
【図 11】



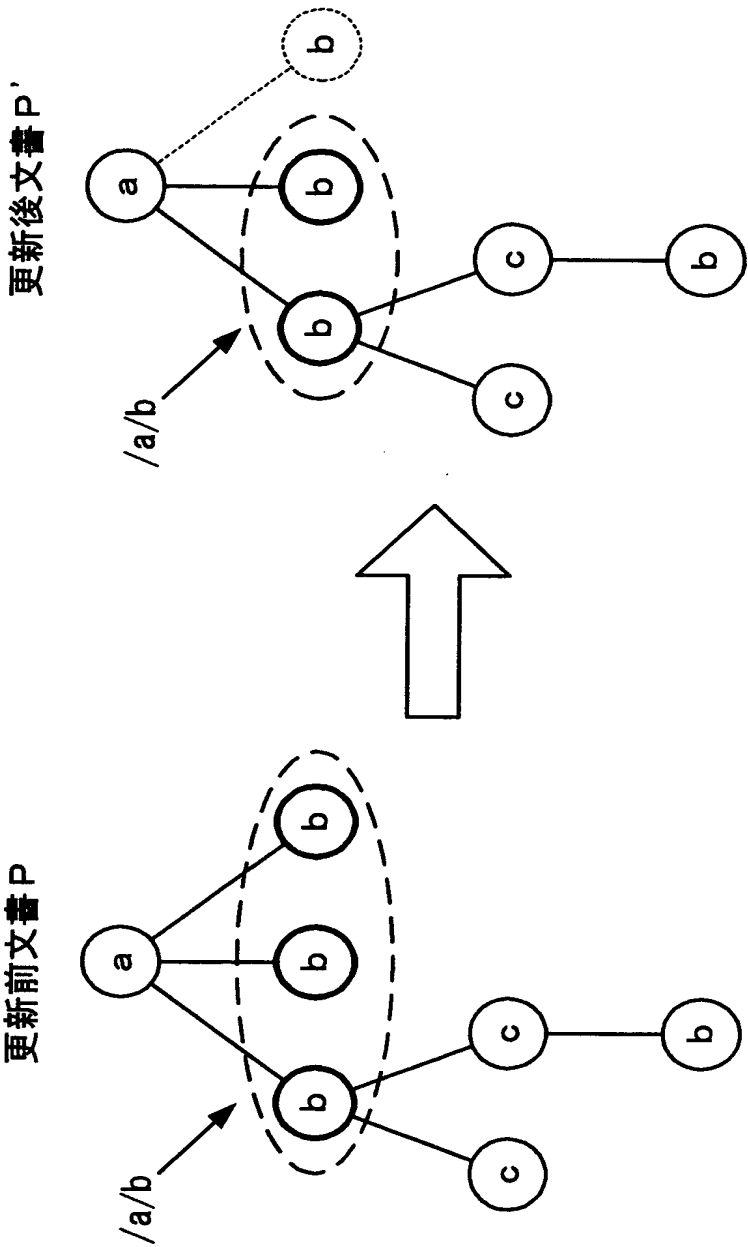
【図 12】



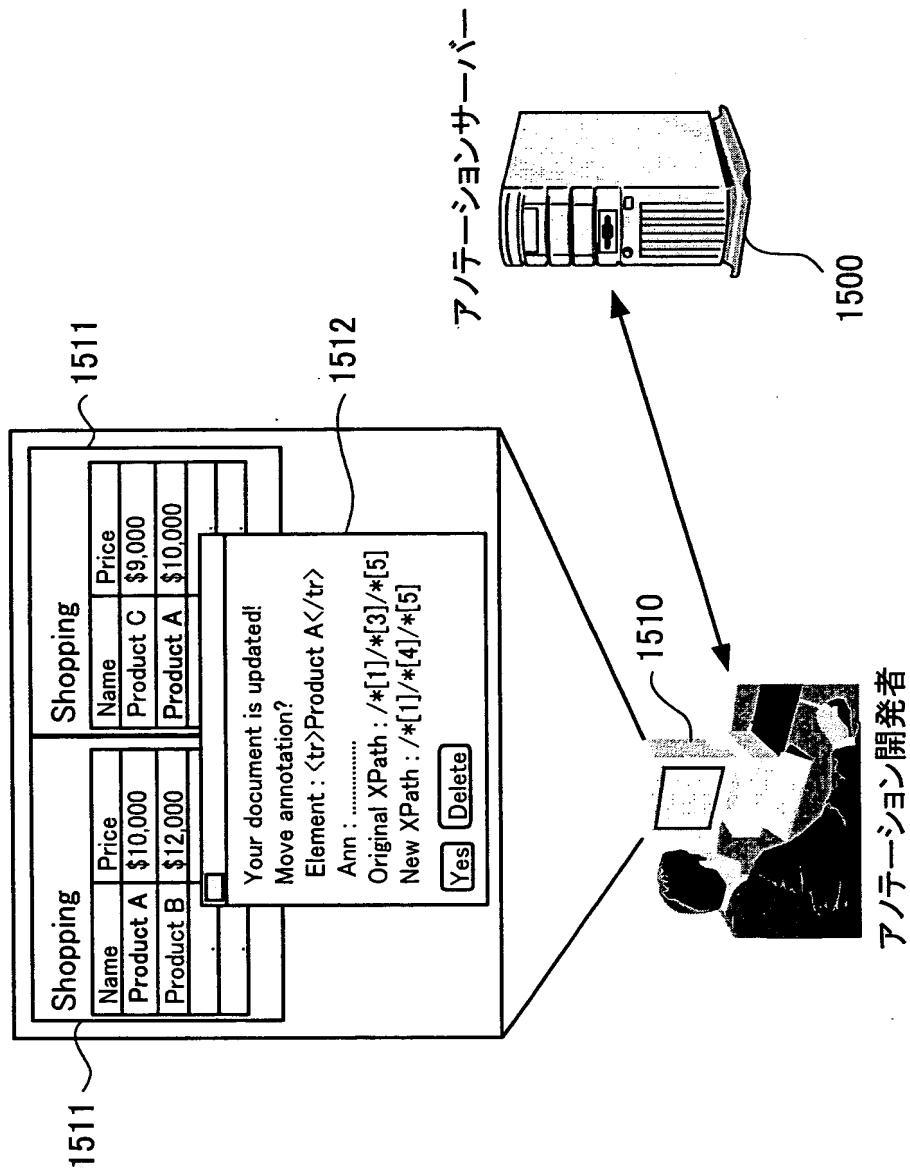
【 図 1 3 】



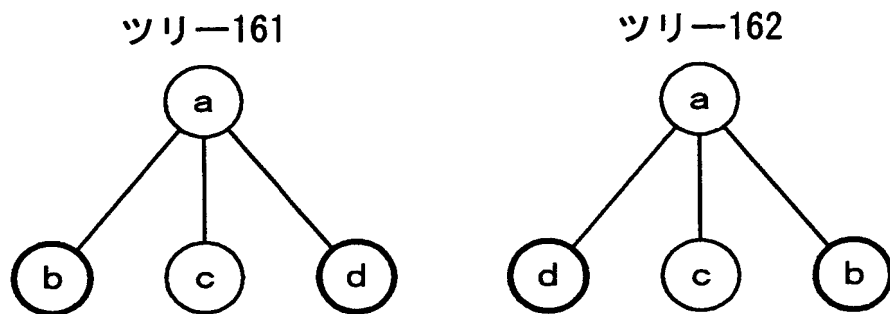
【図 14】



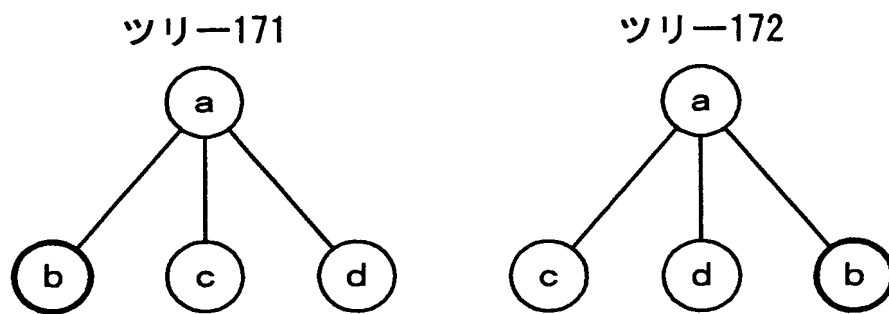
【図 15】



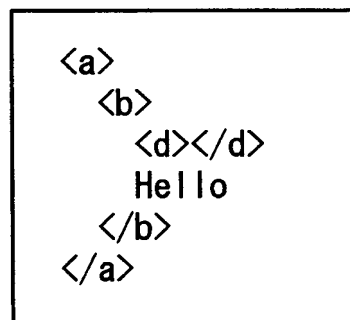
【図 1 6】



【図 1 7】



【図 1 8】



【書類名】 要約書

【要約】

【課題】 構造化文書中の所定の要素が指示されている場合に、この構造化文書が変更されても所望の要素を適切に指示し続けることができるようにする。

【解決手段】 複数の構造化文書の差分を計算する差分計算部 2 0 と、この差分計算部 2 0 により計算された差分に関する情報に基づいて所定の構造化文書における所定の部分を指示する指示情報から他の構造化文書中の対応する部分を指示する指示情報を生成する X P a t h 更新部 3 0 とを備える。

【選択図】 図 2

認定・付加情報

特許出願の番号	特願 2002-206202
受付番号	50201036733
書類名	特許願
担当官	小野寺 光子 1721
作成日	平成14年10月 3日

<認定情報・付加情報>

【特許出願人】

【識別番号】	390009531
【住所又は居所】	アメリカ合衆国10504、ニューヨーク州 アーモンク ニュー オーチャード ロード
【氏名又は名称】	インターナショナル・ビジネス・マシーンス・コーポレーション

【代理人】

【識別番号】	100086243
【住所又は居所】	神奈川県大和市下鶴間1623番地14 日本アイ・ビー・エム株式会社 大和事業所内
【氏名又は名称】	坂口 博

【代理人】

【識別番号】	100091568
【住所又は居所】	神奈川県大和市下鶴間1623番地14 日本アイ・ビー・エム株式会社 大和事業所内
【氏名又は名称】	市位 嘉宏

【代理人】

【識別番号】	100108501
【住所又は居所】	神奈川県大和市下鶴間1623番14 日本アイ・ビー・エム株式会社 知的所有権
【氏名又は名称】	上野 剛史

【復代理人】

【識別番号】	100104880
【住所又は居所】	東京都港区赤坂5-4-11 山口建設第2ビル 6F セリオ国際特許事務所
【氏名又は名称】	古部 次郎

出 願 人 履 歴 情 報

識別番号 [390009531]

1. 変更年月日 2002年 6月 3日

[変更理由] 住所変更

住 所 アメリカ合衆国10504、ニューヨーク州 アーモンク ニ
ュー オーチャード ロード

氏 名 インターナショナル・ビジネス・マシーンズ・コーポレーショ
ン